This is the preprint of the paper **"Change Detection in Moving-Camera Videos with Limited Samples Using Twin-CNN Features and Learnable Morphological Operations"**.

The Published journal article version is available at https://www.sciencedirect.com/science/article/abs/pii/S0923596523000516.

```
@article{PADILLA2023116969,
title = {Change detection in moving-camera videos with limited samples using twin-CNN features and learnable morphological operations},
journal = {Signal Processing: Image Communication},
volume = {115},
pages = {116969},
year = {2023},
issn = {0923-5965},
doi = {https://doi.org/10.1016/j.image.2023.116969},
url = {https://www.sciencedirect.com/science/article/pii/S0923596523000516},
author = {Rafael Padilla and Allan F. {da Silva} and Eduardo A.B. {da Silva} and Sergio L. Netto},
}
```

Change Detection in Moving-Camera Videos with Limited Samples Using Twin-CNN Features and Learnable Morphological Operations

Rafael Padilla, Allan F. da Silva[1,*], Eduardo A. B. da Silva, Sergio L. Netto

Electrical Engineering Program, COPPE/Federal University of Rio de Janeiro, 21945-907, RJ, Brazil.

Abstract

This paper presents a new system to detect changes between reference and target videos suitable for small-scale datasets. Twin pre-trained ResNet-50 features are processed using a learning-based pipeline that has a limited number of adjustable parameters, allowing end-to-end training even on relatively small databases. This is achieved with two innovative modules in tandem: a low-complexity dissimilarity module and a post-processing step using learnable morphological operations. Both can be smoothly incorporated in optimization procedures that employ gradient-based algorithms. The pipeline ends with temporal consistency and change classification modules, and it is evaluated on the VDAO dataset, a challenging database of videos recorded with moving cameras in a cluttered industrial environment. Ablation studies show how each proposed module contributes to the final system performance, with a prominence role for the newly proposed ones. Results indicate that the proposed system achieves a detection performance that is about 18% superior to the one of current state-of-the-art methods. Software, results, and a pre-trained architecture of the proposed framework are available at `https://github.com/rafaelpadilla/TCF-LMO`.

*Corresponding author

Email addresses: `rafael.padilla@smt.ufrj.br` (Rafael Padilla), `allan.freitas@smt.ufrj.br` (Allan F. da Silva), `eduardo@smt.ufrj.br` (Eduardo A. B. da Silva), `sergioln@smt.ufrj.br` (Sergio L. Netto)

[1]Present address: Institute of Systems and Robotics, University of Coimbra, 3030-194 Coimbra, Portugal

Keywords: change detection, moving camera, convolutional networks, learnable morphological operations, limited datasets

1. Introduction

Video-based change detection systems address the problem of finding visual patterns absent from a model of the background [1] or from previous videos [2]. The changes one is looking for are context dependent and part of the system specifications, such as, for example, abandoned luggage in crowded spaces [3] or highway crossing by pedestrians [4]. In industrial facilities, changes in surveillance videos corresponding to materials left in areas that may cause accidents or corresponding to items capable of producing flames are critical, as they could lead to severe consequences if not detected in due time [5].

A very challenging scenario in this context is the change detection problem using videos obtained from a moving camera [6–8]. In such cases, change detection can be performed by comparing a given input video to another reference video representing normal environmental conditions [7–12]. A prominent dataset designed for this scenario is the video database of abandoned objects (VDAO) [5], a publicly available dataset which includes reference (anomaly-free) and target videos (containing abandoned objects that act as the content change) recorded in a cluttered industrial environment. Bounding-box annotations of the abandoned objects in the target videos are also provided, allowing one to train and also to assess several change-detection algorithms [10, 11, 13–17]. Notwithstanding the different techniques employed, all these methods follow a similar processing pipeline: (i) reference and target videos are time-aligned so that their corresponding frames can be readily compared; (ii) the dissimilarity between reference and target videos is computed in some particular domain; (iii) the pixels or frames with a high enough dissimilarity score are associated to a content change.

However, a constraint for such scenario is the limited amount of variability in the data. Consider for instance the VDAO dataset. It contains videos recorded in a fixed environment, with static abandoned objects that only appear for a short period of time, while the camera undergoes a predefined path. Under such conditions, one might not obtain enough diversified samples from each object to train modern classifiers [18, 19].

This work proposes a new method, based on the processing pipeline above, to perform change detection between reference and target videos ac-

quired by a moving camera. The proposed system has a limited number of learnable parameters, thus allowing an effective training procedure even using constrained datasets. In brief, we propose to use concatenated layer-3 ResNet-50 feature tensors from reference and target videos to feed a novel low-complexity dissimilarity module, that was developed to be more flexible than a simple Euclidian distance while also allowing training on small-scale datasets. We also introduce a mathematical morphology module that uses novel approximations of opening and closing operations, where the structuring element sizes can be learned using optimization procedures based on gradient-descent algorithms. To the authors' best knowledge, morphological operations with such characteristics are unprecedented in the literature. The proposed system also incorporates a temporal consistency module to mitigate both false and missed detections. Change-detection results are provided at frame and pixel levels, indicating the system capability of detecting and locating the change in both time and space. Ablation studies are described analyzing the contribution of each proposed module to the final system performance. All code and results related to the proposed pipeline are available at `https://github.com/rafaelpadilla/TCF-LMO`.

The remainder of this paper is organized as follows: Section 2 contains an overview of recent methods on the change detection problem and previous studies related to the VDAO dataset. Section 3 presents the newly proposed pipeline, where each module, as well as the loss function employed, is detailed. Section 4 describes the VDAO dataset and Section 5 presents experimental results obtained on this dataset by the proposed architecture in comparison to previous works. Conclusions are drawn in Section 6 emphasizing the paper main contributions.

2. Literature Review

The goal of change detection algorithms is, given an input video, to recognize the precise moment or position of a new event in the scene, such as an action or object. According to Mandal et al. [20], the goal of change detection techniques is to identify different regions in a video frame given multiple frames of the same scene. With respect to video capture, surveillance systems can be classified into two types: systems using fixed cameras and systems using moving cameras. The first category includes static cameras or pan-tilt-zoom (PTZ) cameras, which allow zooming and rotations. In

the second category are moving cameras, that provide a wider visual coverage of the environment with a reduced number of cameras.

When employing static cameras, background subtraction has been widely applied to detect anomalies or changes [21–23]. Some works apply static cameras to detect moving objects, and often allow a small amount of movement usually due to some kind of camera jitter [6, 24–27].

In a more challenging scenario, works such as [10–12, 28, 29] employ moving cameras to supervise larger areas, at the expense of an increase in the complexity of the associated task. In fact, as the camera moves, so do shadows, occlusions, and light reflections, which limit the success of techniques designed for static cameras [7, 17, 30].

The system described by Kong et al. [10] was able to detect abandoned objects along a road by mounting a camera on a moving car. Global positioning system (GPS) coordinates were used to align the reference and target frames and a homography was used to estimate affine transformations that register the corresponding frames. Similarly, Mukojima et al. [11] perform change detection along train tracks by recording videos with a camera mounted on a train. After aligning reference and target videos, different similarity measurements are applied to detect the presence of anomalies in the scenes.

As discussed in Section 1, the VDAO database [5] presents videos obtained with a moving camera in a cluttered industrial environment. Reference videos represent normal situations and target videos contain abandoned objects representing anomalous situations. Different works have attempted to identify the content change between reference and target videos, both at the frame level, classifying the frames as anomalous or non-anomalous, or at the object or pixel level, by identifying the positions of the content change in the target frame.

The work of Nakahata et al. [13] proposes a modified spatio-temporal composition (STC) algorithm [31] to mitigate false detections caused by the camera movement and applies its solution on the VDAO and UCSD databases. The moving-camera STC breaks down the videos into smal spatio-temporal volumes that are represented by codewords from a codebook, the ones with lower probability values being associated to significant video changes. Carvalho et al. [16] employ a multi-scale video analysis to compare reference and target videos, registered frame-by-frame using SURF features, and, for each scale, anomalous regions containing abandoned objects are identified by a normalized cross-correlation (NCC) operation. Inspired by the low-rank sub-

space decomposition of Bian et al. [32], the work of Thomaz et al. [14] uses the reference video to generate a low-rank subspace, and the target videos are decomposed into a low-rank component plus a residual corresponding to the camera jitter and an innovation corresponding to an abandoned object not present in the reference video. The detection results obtained by Thomaz et al. [14] are improved by Jardim et al. [17], that integrate to the detection pipeline a two-stage optimization process, where the inner loop estimates the best geometric transformation between target and reference videos, and the outer loop estimates the best decomposition in a low-rank video, a residual, and the innovation corresponding to the abandoned object.

Following a different line from the above works, Afonso et al. [15] propose a learning-based system to detect the abandoned objects in the VDAO. In it, the ResNet-50 convolutional layers [33] are used as feature extractors from both reference and target frames. The mere concatenation of these features, however, highly increases the problem dimension thus hindering the classifier development on data-limited datasets. For this reason, the Euclidean distance between the reference and target feature tensors is employed with random forest or multilayer perceptron classifiers to detect possible changes within the target video.

The present work also uses a learning-based pipeline but incorporates several innovations to the framework employed in previous works. In particular, the newly proposed dissimilarity module (DM) combines pretrained ResNet-50 features in a more flexible manner. It was developed to allow the model to explore the feature information more efficiently than the Euclidean distance while possessing a sufficiently low number of parameters that makes it possible to train the model on small-scale datasets using gradient-based algorithms. To increase system robustness, post-processing steps are implemented enforcing temporal and spatial consistency in the detected changes. However, different from previous works [13, 14, 16, 17], the new temporal consistency module (TCM) is entirely integrated into the optimization procedure and does not require hyperparameter tuning. Finally, the new morphology module (MM) performs novel operations that approximate the ones from traditional mathematical morphology while also allowing for gradient-based learning, which once more obviates the tuning of hyperparameters.

3. The TCF-LMO Change Detection System

The proposed change detection framework, referred to as TCF-LMO (Twin Convolutional Features and Learnable Morphological Operations) system, is shown in Figure 1. The function and implementation of each module are detailed in the subsections that follow.

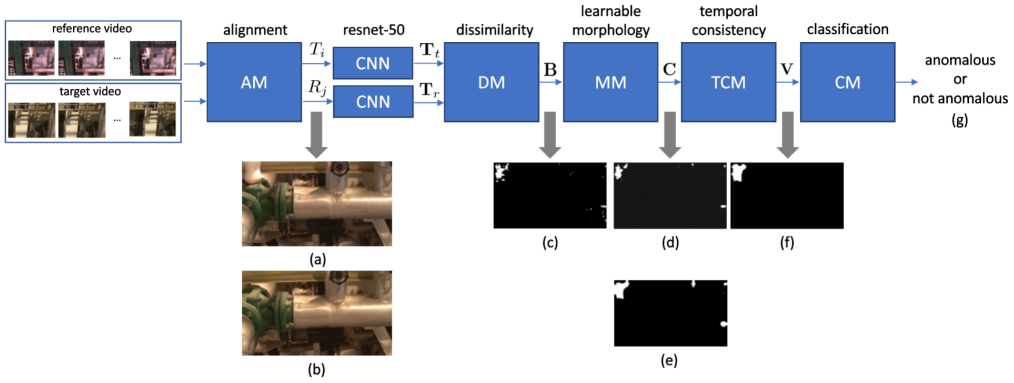


Figure 1: Block diagram of the proposed pipeline. A pre-processing module is used to align a pair of target and reference videos. Given an aligned pair (T_i, R_j) of target (a) and reference (b) frames, the ResNet-50 network extracts the corresponding $\mathbf{T_t}$ and $\mathbf{T_r}$ feature tensors, respectively. The dissimilarity module generates a single binary image $\mathbf{B}$ that represents the pixel-level dissimilarity between the input tensors $\mathbf{T_t}$ and $\mathbf{T_r}$. The morphological opening and closing operations are then used to remove and connect isolated detections, resulting in images (d) and (e), respectively. Temporal consistency is employed to remove false positive pixels from (e) that do not persist in consecutive frames, resulting in image (f). Based on the number of active pixels in (f), the final module outputs the classification (g) of the target frame as anomalous or not.

3.1. Alignment Module

The alignment module (AM) performs a temporal alignment based on a fixed-length temporal sliding window that moves along the frames of the reference signal. Given a target video T, we prealign its initial frame T_0 to its best frame correspondence in the reference video R, creating an aligned pair (T_0, R_{j_0}). For a target frame T_{i+1}, we use the previously aligned pair (T_i, R_{j_i}) and compute the similarity, measured with the Frobenius distance, between T_{i+1} and each reference frame R_{j_i-5}, R_{j_i+5} within a search window of 11 frames. The reference frame with the lowest distance to T_{i+1} defines a new aligned pair $(T_{i+1}, R_{j_{i+1}})$.

With this method, one is able able to align the current target frame to the reference video as soon as it is available, allowing the computation to be performed on-the-fly. In practice, this alignment may compensate for undesired movements of the robot and camera jitters during the video acquisition.

3.2. Feature Extraction Module

Two convolutional layers of the ResNet-50 network [33] are used as feature extractors for both reference and target videos. Afonso et al. [15] extract features from the 3rd residual layer and show that they provide the best anomaly discrimination results, probably due to their good compromise between low-level and high-level characterization of the image contents. Following that work, and to allow a direct comparison to it, the outputs from that same layer were employed in the current system as feature tensors. The 3rd layer of a pre-trained ResNet-50, that is kept frozen during the training, is used to extract features from an aligned pair of target and reference frames, so that the aligned pair of frames $(T_{i+1}, R_{j_{i+1}})$ is transformed into the corresponding pair of feature tensors $(\mathbf{T}_t, \mathbf{T}_r)$, where $\mathbf{T}_t = [\mathbf{T}_{t_1}, ..., \mathbf{T}_{t_{256}}]$ and $\mathbf{T}_r = [\mathbf{T}_{r_1}, ..., \mathbf{T}_{r_{256}}]$, with $\mathbf{T}_{t_i} \in \mathbb{R}^{90\times160}$ and $\mathbf{T}_{r_i} \in \mathbb{R}^{90\times160}$. In brief, the AM provides a pair of reference and target frames with 360×640 pixels each. When processed by the ResNet-50, each frame results in a feature tensor equivalent to a total of $C = 256$ feature maps of dimensions 90×160 each.

3.3. Dissimilarity Module

Given the feature tensors of a synchronized pair of reference and target frames, the dissimilarity module (DM) produces a binary image whose white pixels represent the most contrasting regions of both frames, which are likely to represent a significant change. For instance, Afonso et al. [15] produce such binary image by directly subtracting the feature tensors and inputting this difference to a random forest classifier. Ideally, the DM would have increased detection performance if instead of such a difference its input was the information-richer concatenation of the features from the reference and target frames. However, as their concatenation has twice as much samples as their difference, the DM input would have twice the dimension, which would tend to significantly increase the DM number of trainable parameters. Since the more trainable parameters one has the larger has to be the amount of data used for training, such module based on feature tensor concatenation would not be suited for data-deficient applications.

In this work, we propose a novel low-complexity DM architecture that possesses a sufficiently small number of parameters so that it can be trained in relatively limited databases, while also being able to take full advantage of the wealth of information provided by the feature concatenation. Differently from approaches using attention mechanisms and squeeze-and-excitation networks [34–37], which emphasize channels output by convolutional layers and require the training of ∼2.5 million parameters [34], our approach also weights the feature maps but involves only a reduced number of parameters, as convolutional layers are avoided. Assuming that C is the number of feature maps extracted by the convolutional network, our architecture has $(4C + 1)$ learnable parameters. In particular, as described in Subsection 3.2, the ResNet-50 3rd residual layer outputs $C = 256$ feature maps, which correspond to only 1025 adjustable parameters.

This is achieved by considering an array of learnable weights to combine each pair of reference/target feature maps. For that, we initially adjust a total of $2C = 512$ weights, being $C = 256$ weights w_{r_i} for the reference feature maps $\mathbf{T}_{r_i}$ and $C = 256$ weights w_{t_i} for the target feature maps $\mathbf{T}_{t_i}$, with $i = 1, 2, \ldots, C$. For each channel i the weighted feature tensors are subtracted element-wise and added a bias component b_i. The application of the tanh function generates the $\mathbf{T}_i$ maps, that are thus given by

$$\mathbf{T}_i = \tanh\left(w_{r_i}\mathbf{T}_{r_i} - w_{t_i}\mathbf{T}_{t_i} + b_i\mathbf{1}\right), \tag{1}$$

where $\mathbf{1}$ denotes a matrix with the same dimensions as $\mathbf{T}_i$ with all elements equal to one. Each $\mathbf{T}_i$ map is then multiplied by another adjustable weight w_i and added altogether along the channel axis to generate a single and final feature map with resolution 90×160. To generate a binary-like image, the final feature map is subtracted by a value t and soft-thresholded with a sigmoid function of the form

$$s(x) = \frac{1}{1 + e^{-\gamma x}}, \tag{2}$$

where γ is a hyperparameter that controls the soft threshold derivative.

Therefore, the DM output image $\mathbf{B}$ is obtained as

$$\mathbf{B} = s\left(\left(\sum_{i=1}^{C} w_i \mathbf{T}_i\right) - t\mathbf{1}\right), \tag{3}$$

with $\mathbf{1}$ and $\mathbf{T}_i$ as defined in Eq. (1). We empirically set $\gamma = 5$ during the training stage, which is enough to avoid gradient divergence during the gradient-descent training process.

Eqs. (1) to (3) imply that the DM is built with a set of operations widely used by neural networks, thus being naturally trainable following a gradient-based framework. The entire DM pipeline is illustrated in Figure 2, where the blocks represent the tensors and the text in red highlights the operations performed in every module step. From this representation, one clearly notices a total of $3C$ weights (w_{r_i}, w_{t_i}, and w_i), C bias parameters b_i, and a threshold t, totalling $(4C + 1) = 1{,}025$ learnable parameters.

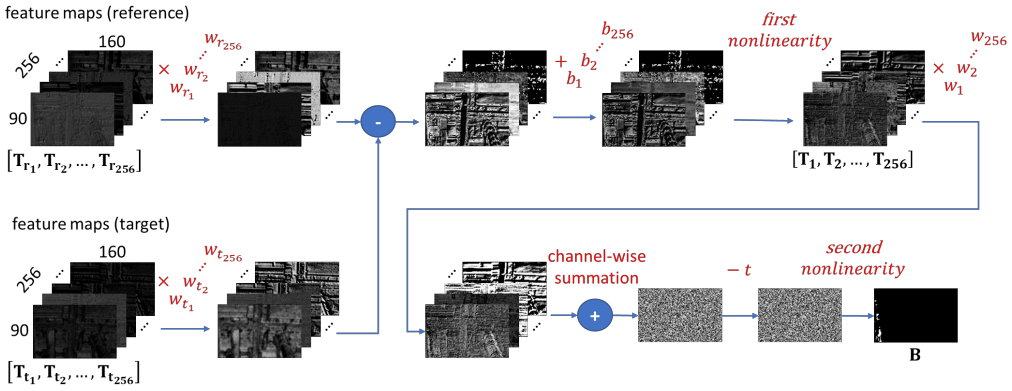


Figure 2: Complete schematic for the DM representing all operations and 1,025 adjustable parameters employed to produce a binary image $\mathbf{B}$.

From Eqs. (1), (2), and (3), the partial derivatives of the binary image $\mathbf{B}$ with respect to each adjustable parameter can be computed as

$$\frac{\partial \mathbf{B}}{\partial t} = -\gamma\, \mathbf{B} \odot (\mathbf{1} - \mathbf{B}), \tag{4}$$

$$\frac{\partial \mathbf{B}}{\partial w_i} = \gamma\, \mathbf{T_i} \odot \mathbf{B} \odot (\mathbf{1} - \mathbf{B}), \tag{5}$$

$$\frac{\partial \mathbf{B}}{\partial b_i} = \gamma w_i \mathbf{B} \odot (\mathbf{1} - \mathbf{B}) \odot (\mathbf{1} - \mathbf{T_i} \odot \mathbf{T_i}), \tag{6}$$

$$\frac{\partial \mathbf{B}}{\partial w_{r_i}} = \gamma w_i \mathbf{B} \odot (\mathbf{1} - \mathbf{B}) \odot \mathbf{T_{r_i}} \odot (\mathbf{1} - \mathbf{T_i} \odot \mathbf{T_i}), \tag{7}$$

$$\frac{\partial \mathbf{B}}{\partial w_{t_i}} = -\gamma w_i \mathbf{B} \odot (\mathbf{1} - \mathbf{B}) \odot \mathbf{T_{t_i}} \odot (\mathbf{1} - \mathbf{T_i} \odot \mathbf{T_i}), \tag{8}$$

where $\odot$ represents the Hadamard point-wise product and matrix $\mathbf{1}$ is defined as before.

In the proposed system, the parameter adjustment, as detailed in Algorithm 1 below, is performed by optimizing a loss function based on the Matthews correlation coefficient (MCC) with a few adaptations as discussed in Subsection 3.7.

Algorithm 1: **Training the DM**

1. *Inputs*: N batches of 14 balanced data frame pairs $(\mathbf{T_t}, \mathbf{T_r})$.
2. *Outputs*: Optimal parameters w_{r_i}, w_{t_i}, b_i, w_i, and t, for $i = 1, 2, \ldots, 256$.
3. *Initialization*: Set $w_{r_i} = w_{t_i} = w_i = 1$ and $b_i = 1 \times 10^{-2}$, for $i = 1, 2, \ldots, 256$.
4. *Training steps*:

   For each input batch:

   For each frame pair in the batch:

   i. Obtain binary ground-truth image $\mathbf{G}$;

   ii. Compute dissimilarity output $\mathbf{B}$ with Eq. (3);

iii. Compute MCC between **B** and **G** with Eq. (22);

end;

iv. Compute batch loss MSE with Eq. (23);

v. Compute local gradients with Eqs. (4) to (8);

vi. Apply the chain rule to update parameters using learning rates from Table 3;

end;

vii. Terminate procedure.

3.4. Learnable Morphology Module

In a machine learning framework, mathematical morphology operations have been successfully applied as a pre- [38] or post-processing step [39, 40]. To improve the performance of change detection algorithms, a morphological opening can be applied to remove isolated false positive pixels and a subsequent closing operation can be used to fill possible gaps in the resulting blobs. As the number of scattered pixels and the sizes of the blobs may vary, the radius of the structuring elements in both opening and closing operations must be adjusted during system training and validation in order to post-process the binary image in an optimal way. Unfortunately, these operations involve a combination of functions, such as minima and maxima over sets, that cannot be directly incorporated in a gradient-based learning framework of a neural network (would allow their parameters to be learnt). For that purpose, we propose in the morphology module (MM) a novel structure that implements morphological-like operations that enable the optimization of the structuring element size, allowing these parameters to be trained together with the other system parameters.

Given an image $f(x, y)$, the standard erosion operation identifies the (x, y)-plane regions where the structuring element $h(x, y)$ is totally contained within the image regions where $f(x, y) = 1$. One can then approximate the erosion operation by a convolution $g(x, y)$ between the image $f(x, y)$ and the structuring element $h(x, y)$, followed by a hard thresholding function $v(x, y)$. In this approximation, whenever $h(x, y)$ is entirely contained in an active portion of the image $f(x, y)$, the convolution $g(x, y) = f(x, y) * h(x, y)$ shall output the structuring element area A, and the thresholding function

$$v(x, y) = \begin{cases} 0, & \text{if } g(x, y) < A, \\ 1, & \text{otherwise,} \end{cases} \tag{9}$$

can be used to identify these situations. Following this framework, a learnable erosion operation can be obtained if both the structuring element $h(x, y)$ and the thresholding function $v(x, y)$ are composed of functions that can be integrated in a learning pipeline. This can be achieved by using the operations

$$h(x, y) = 1 - s\left(\sqrt{x^2 + y^2} - R\right), \tag{10}$$

$$v(x, y) = s\big(g(x, y) - (A - \epsilon)\big), \tag{11}$$

where R is the radius of the circular structuring element $h(x, y)$ of area $A = \pi R^2$, $s(\cdot)$ is the sigmoid function (with $\gamma = 5$) defined in Eq. (2), and ϵ is a tolerance margin for the threshold value of $v(x, y)$ empirically set to $\epsilon = 0.08A$. With the definitions above $h(x, y)$, $g(x, y)$, and $v(x, y)$ have the following derivatives with respect to R:

$$\frac{\partial h(x, y)}{\partial R} = \gamma h(x, y)(1 - h(x, y)), \tag{12}$$

$$\frac{\partial g(x, y)}{\partial R} = f(x, y) * \frac{\partial h(x, y)}{\partial R}, \tag{13}$$

$$\frac{\partial v(x, y)}{\partial R} = \gamma v(x, y)(1 - v(x, y))\left(\frac{\partial g(x, y)}{\partial R} - 2kR\right), \tag{14}$$

respectively, with $kR^2 = A - \epsilon$.

Following the erosion/dilation duality, one can express the dilation of the image $f(x, y)$ with the structuring element $h(x, y)$ by computing the complement image of the erosion operation between the complemented $f(x, y)$ with $h(x, y)$. Therefore, the erosion and dilation operations of image $f(x, y)$ with the structuring element $h(x, y)$ can be respectively defined as

$$f(x, y) \ominus h(x, y) = s\big(g(x, y) - (A - \epsilon)\big), \tag{15}$$

$$f(x, y) \oplus h(x, y) = 1 - s(1 - g(x, y) - (A - \epsilon)), \tag{16}$$

with $g(x, y) = f(x, y) * h(x, y)$, as before.

A similar development can be used, for instance, to define learnable opening and closing operations as a composition of the erosion and dilation operations in direct and reverse orders, as given by

$$f(x, y) \circ h(x, y) = \big(f(x, y) \ominus h(x, y)\big) \oplus h(x, y), \tag{17}$$

$$f(x, y) \bullet h(x, y) = \big(f(x, y) \oplus h(x, y)\big) \ominus h(x, y). \tag{18}$$

The steps to train the MM are detailed in Algorithm 2.

Algorithm 2: **Training the MM**

1. *Inputs*: N batches of 15 balanced data frame pairs $(\mathbf{T_t}, \mathbf{T_r})$.
2. *Outputs*: Optimal parameters R_o and R_c.
3. *Initialization*: Set $R_\mathrm{o} = R_\mathrm{c} = 1$.
4. *Training steps*:

   i. Freeze gradients of all modules except MM;

   For each input batch:

   For each frame pair in the batch:

   ii. Obtain binary ground-truth image $\mathbf{G}$;

   iii. Compute dissimilarity output $\mathbf{B}$ with Eq. (3);

   iv. Create $h_o(x, y)$ with radius R_o using Eq. (10);

   v. Create $h_c(x, y)$ with radius R_c using Eq. (10);

   vi. Compute $\mathbf{O} = \mathbf{B} \circ h_o(x, y)$ using Eq. (17);

   vii. Compute $\mathbf{C} = \mathbf{O} \bullet h_c(x, y)$ using Eq. (18);

   viii. Compute MCC between $\mathbf{C}$ and $\mathbf{G}$ with Eq. (22);

   end;

   ix. Compute batch loss MSE with Eq. (23);

   x. Compute local gradients with Eq. (12) to (14);

   xi. Apply the chain rule to update parameters using learning rates from Table 3;

   end;

   xii. Terminate procedure.

3.5. Temporal Consistency Module

Although the MM is able to produce binary images that reveal most of the structural differences between the reference and target frames, its output is highly affected by differences caused by variable light reflections, misalignment, and shadows in consecutive frames. The next module in the

proposed pipeline, the temporal consistency module (TCM), mitigates these effects by removing false positive pixels that are not consistent in consecutive frames.

For that matter, a temporal voting window is employed on the sequence of binary input images $\mathbf{C}$ obtained from the MM (see Algorithm 2). Consider a window $W_{c,\ell} = [\mathbf{C}_{c-k}, \ldots, \mathbf{C}_c, \ldots, \mathbf{C}_{c+k}]$ of length $\ell = 2k + 1$ frames centered at time instant c. Similarly to [14–17], we adopt a criterion that keeps a positive (white) pixel at position (x, y) only if the majority of the corresponding pixels $[\mathbf{C}_{c-k}(x, y), \ldots, \mathbf{C}_{c+k}(x, y)]$ within the temporal window $W_{c,\ell}$ are also positive, generating the pixel (x, y) of the binary output image $\mathbf{V}$. We estimate the value of ℓ by evaluating different voting window lengths within the interval $1 \leq \ell \leq 15$, and choosing the one that minimizes the MCC (see Subsection 3.7) between the resulting image $\mathbf{V}$ and the corresponding ground truth $\mathbf{G}$.

3.6. Classification Module

The classification module (CM) is the final stage in the proposed pipeline, as illustrated in Figure 1. The CM receives an image $\mathbf{V}$ with blobs, corresponding to the possible content changes in the target frame, and determines whether or not this frame contains an anomaly based on the active portion of the image. To perform this operation, the number of anomalous pixels is counted by summing up all active pixels in the image, and its difference from a threshold value t_{CM} is input to a sigmoid function as given in Eq. (2), with $\gamma = 5000$ obtained empirically. This yields the final pixel-wise classification as given by

$$Y = s\big(v - t_{\mathrm{CM}}\big), \tag{19}$$

where v is the total number of active pixels in image $\mathbf{V}$. If $Y > 0.5$, a significant change is identified in the input frame; otherwise no change is detected. The threshold t_{CM} is learned using gradient-descent. The derivative of Y with respect to t_{CM} is given by

$$\frac{\partial Y}{\partial t_{\mathrm{CM}}} = -\gamma\, Y(1 - Y). \tag{20}$$

The value of t_{CM} is adjusted following a gradient-based procedure in order to minimize the classification error of the proposed pipeline according to Algorithm 3.

Algorithm 3: **Training the CM**

1. *Inputs*: N batches of 14 sequential data frame pairs $(\mathbf{T_t}, \mathbf{T_r})$.
2. *Outputs*: Optimal parameter t_{CM}.
3. *Initialization*: Set $t_{\mathrm{CM}} = 2 \times 10^{-2}$.
4. *Training steps*:
   i. Freeze gradients of all modules except CM;

   For each input batch:

   ii. Classify all images $\mathbf{V}$ produced with the input batch producing their corresponding Y values;

   iii. Compute MSE between output Y and corresponding ground-truth values;

   iv. Compute gradient with Eq. (20);

   v. Apply the chain rule to update t_{CM} using the learning rate from Table III;

   end;

   vi. Terminate procedure.

3.7. Loss Function: Modified Matthews Correlation Coefficient

The DM, MM, and TCM outputs are binary-like images whose pixels with large values (white) represent regions of the target frame that are substantially different from the reference frame. Therefore, the connected white pixels (blobs) in these images are expected to represent the silhouettes of anomalous objects. The current version of the VDAO database contains only rectangular ground-truth annotations representing the anomalous objects.

A proper training of the proposed pipeline modules requires a metric to compare how close the binary output image is to the binary ground truth. In a pixel-level comparison of two binary images, the first and straightforward metric used to measure the difference between the two images is the mean-squared error (MSE). However, as the numbers of ground truth white (anomalous) and black (not anomalous) pixels are unbalanced, the MSE when used as a loss function leads to unsatisfactory results [41–44]. For this

reason, we have adopted the Matthews correlation coefficient (MCC) as a loss function to express the differences between the ground-truth and the output images [41–43].

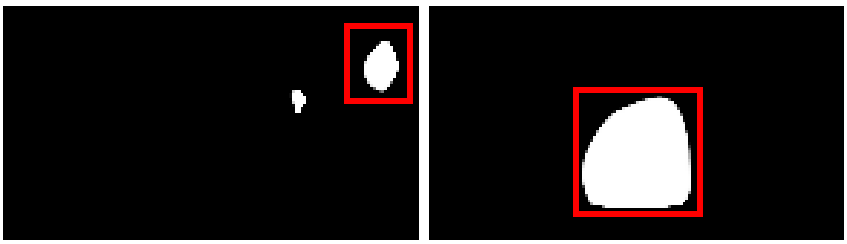
The MCC can be determined by

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}, \tag{21}$$

where TP, FP, TN, and FN are the numbers of true positive, false positive, true negative, and false negative detections, respectively, at the pixel level of the output image in comparison with the ground-truth image. In the best-expected case, FN = 0 and FP = 0, which result in MCC = 1. In the worst case, TP = 0 and TN = 0, which lead to MCC = −1. When the probability of a positive detection is equal to the proportion of the positive samples of the ground-truth [44], one has MCC = 0. Examples of the comparison between output and ground truth images using the MCC metric are shown in Figure 3.

The MCC metric as described by Eq. (21), however, has a critical problem that might prevent it from being used as a loss function in degenerate cases. If one compares two images where at least one of them is constant (containing either only background or foreground), at least one of the factors in the denominator of Eq. (21) becomes null, leading to an undefined MCC value. These constant-image cases are reasonably likely to happen in the VDAO database, as illustrated in Figures 3(c) and 3(d). To circumvent this issue, we propose to make simple modifications in the binary images prior to the computation of the MCC. We reverse a single pixel label (positive or negative) simultaneously in both images, in order to artificially create a pixel containing a TP, FP, TN or FN, depending on the null factor in the MCC denominator.

Let us consider the example shown in Figure 3(c), which contains a bounding box, therefore it contains an anomalous object to be detected, and the network output is black, indicating that no anomaly was detected. For this case, FN = K (size of the bounding box), TN = L (number of pixels outside the bounding box), and TP = FP = 0, resulting in an undefined MCC value as given by Eq. (21). We can adjust the MCC computation by selecting a single pixel outside the bounding box and changing its label in both the detection and ground-truth images, creating an artificial TP pixel. This modification leads to FN = K, TN = $(L - 1)$, TP = 1, FP = 0, and a computable positive MCC value.



(a) MCC = 0.45 (b) MCC = 0.76

(c) MCC = undefined (d) MCC = undefined

Figure 3: Examples of images output by the DM, TCM, and MM stages and their respective MCC scores. The red boxes in (a), (b), and (c) represent ground-truth bounding boxes, where the anomalous objects are located, and the white pixels represent the detected anomalies. The MCC values are obtained according to Eq. (21) and cannot be computed for examples (c) and (d) that present a null denominator.

A similar development can be made for the other degenerate cases, such as the one in Figure 3(d). With this strategy, the MCC can be computed and used as a metric to optimize all DM, MM, and TCM learning parameters, as desired. According to Eq. (21), the best-expected case, where FN = 0 and FP = 0, MCC is 1, and in the worst case, where TP = 0 and TN = 0, MCC is -1. The value of the computed MCC is normalized within the range $[0, 1]$, by making

$$\mathrm{MCC}_{\mathrm{norm}} = \frac{\mathrm{MCC} + 1}{2}, \tag{22}$$

where $\mathrm{MCC}_{\mathrm{norm}} = 1$ is now the best-expected MCC value and $\mathrm{MCC}_{\mathrm{norm}} = 0$ the worst one.

The losses of modules DM, TCM, and MM are computed as an MSE considering MCC of the N samples within a batch and the best-expected

MCC, that is

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{MCC}_{\mathrm{norm}_i} - 1)^2. \tag{23}$$

4. The VDAO Dataset

The VDAO is a challenging database covering anomaly detection in large areas. This database, presented originally by da Silva et al. [5] and available at [45], is composed of videos recorded in a cluttered industrial environment. The 77 videos contained in the VDAO database were recorded by a camera mounted on a *iRobot Roomba*, making a back-and-forth movement on a predefined linear trajectory along a 6-m path. To increase database variability, the VDAO videos were recorded with illumination changes and two different cameras, both with resolution 1280 x 720 pixels and 24 frames per second.

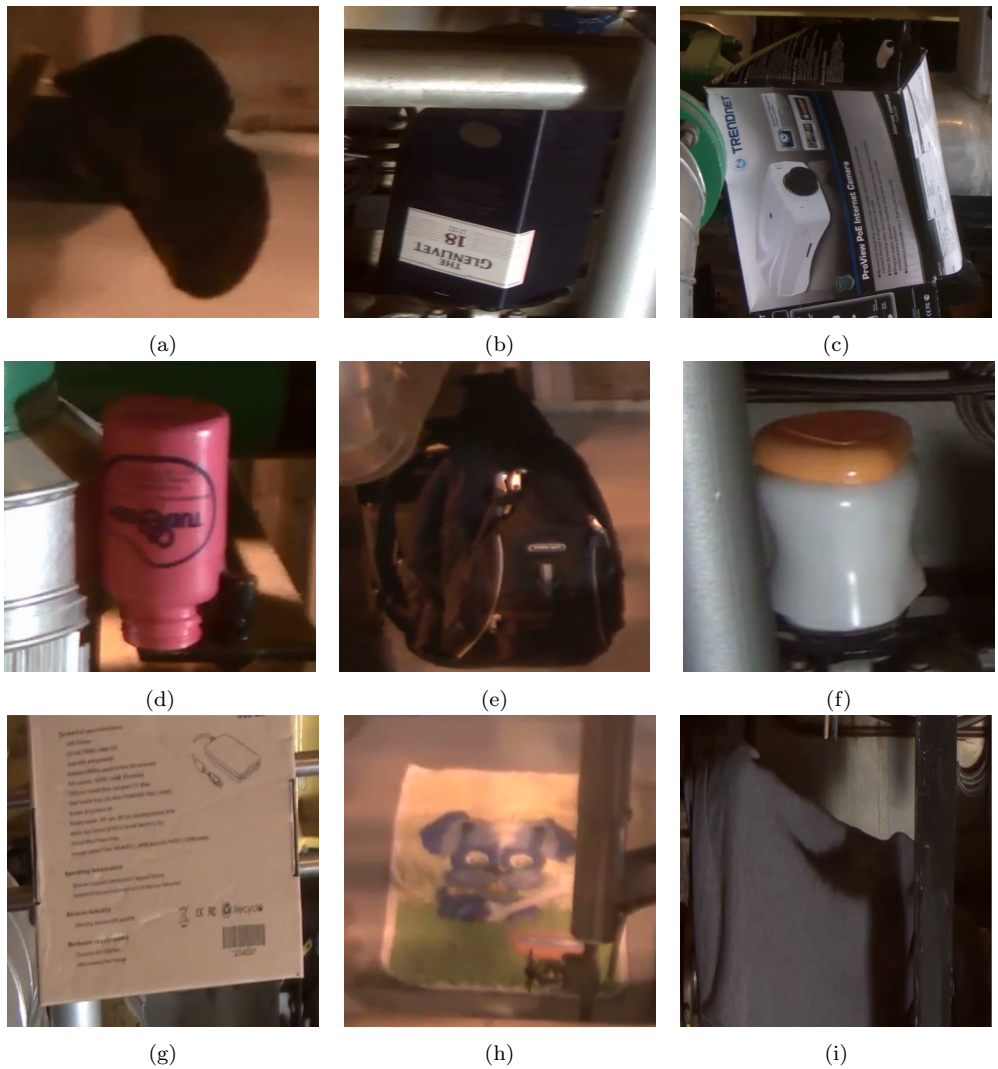
A total of 24 distinct objects were placed in the scenario, simulating objects that do not belong to the environment in normal conditions. Two groups of videos were obtained: the 'reference' videos are those without any anomalous object inserted in the industrial scene and the 'target' videos contain one or more abandoned objects. In the target videos, 15 objects compose videos with multiple objects, and each of the remaining nine objects compose videos with a single object, as illustrated in Figure 4. For each VDAO target video, there is an annotation file containing bounding boxes for the position of each abandoned object in each frame.

4.1. VDAO-200: The Testing Database

In the first works developed with the VDAO database [8, 13, 14], a representative subset of the VDAO was selected as a benchmark for anomaly detection problems. This auxiliary testing database, available at [46], contains 59 single-object video excerpts, with only 200 frames each, with the nine objects shown in Figure 4 in different positions and illumination conditions. The whole VDAO-200 database was designed to have a balanced 50% frame split with or without an abandoned object.

4.2. Assessment Metrics

In order to evaluate the algorithm performances on the VDAO-200, two evaluation scenarios were devised: the frame-level and object-level assessment strategies.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4: Objects used in videos with single objects: (a) shoe (O1); (b) dark blue box (O2); (c) camera box (O3); (d) pink bottle (O4); (e) black backpack (O5); (f) white jar (O6); (g) brown box (O7); (h) towel (O8); (i) black coat (O9).

The frame-level metrics evaluate the system performance in properly classifying each target frame as a whole as anomalous or not. If the frame contains a ground truth bounding box, a TP occurs if the system indicates that it contains an anomalous object, and an FN occurs otherwise. Conversely, if the frame does not contain any ground truth bounding box, an FP occurs if

the system indicates that it contains an anomalous object, and a TN occurs otherwise.

The object-level metrics (also referred to as pixel-level in some works) evaluate the classification of frames based on the intersection of the output image and a ground-truth bounding box, and were created aiming at methods that estimate a pixel-wise map of the anomalies on the target frame. In this approach, each region with connected anomalous pixels is considered an object, and the output image is split into disjoint objects representing the anomalies. A detection mask counts as an object-level TP if it has a non-empty intersection with a ground-truth bounding box, and counts as an object-level FN if the intersection is empty. If any object in the detection mask has an empty intersection with all ground-truth bounding boxes, it is counted as an object-level FP. On the other hand, if there are no ground-truth bounding boxes, a TN is counted if the output detection mask contains no objects, otherwise an FP is also counted. Note that object-level metrics allow a given frame to have simultaneous TPs and FPs or FNs and FPs.

To assess the performance on each VDAO-200 video, all previous works compute the TP and FP rates (TPR and FPR, respectively) for each video and also determine their best compromise in a single DIS value, as given by

$$\mathrm{DIS} = \sqrt{(1 - \mathrm{TPR})^2 + \mathrm{FPR}^2}, \tag{24}$$

which represents the minimum distance of the (TPR, FPR) point to the point of ideal behaviour (TPR = 1 and FPR = 0) on the TPR × FPR plane.

The DIS values are integrated over different videos of the VDAO-200 dataset using two different approaches, namely the average DIS ($\mathrm{DIS_{av}}$) and the overall DIS ($\mathrm{DIS_{oa}}$). The $\mathrm{DIS_{av}}$ is determined by the straightforward average DIS value for all 59 VDAO-200 videos. The $\mathrm{DIS_{oa}}$ contemplates the fact that each VDAO-200 video contains a different amount of anomalous frames, and, therefore, the combined metric weights the individual DIS scores for each video accordingly. The computation of this metric is equivalent to concatenating all 59 videos in a single video and obtaining the total TPR and FPR, allowing one to use Eq. (24) to determine the $\mathrm{DIS_{oa}}$ value.

5. Experimental Results and Analysis

We evaluate the proposed pipeline on the single-object videos of the VDAO dataset. For that, we initially define our procedure to train, validate, and test the proposed pipeline. Afterwards, we compute all metrics

defined in Section 4.2 and compare our results to the ones reported for the DAOMC [10], ADMULT [16], MCBS [11], mcDTSR [17], and CNN+RF [15] algorithms.

5.1. System Development

The training process of the proposed system depicted in Figure 1 uses a validation procedure to evaluate generalization capability and perform hyperparameter selection. The epoch in which the model produces the lowest loss value in the validation set is considered to have the best set of learned parameters, which are then are used on a testing set to evaluate the system performance.

To avoid contamination between training, validation, and test sets, we partition the set of 59 VDAO single-object videos according to which of the 9 objects each video contains, creating sets O_i, $i = 1, \ldots, 9$ (see Figure 4). Nine folds are created such that in fold k the set O_k is the test set and the remaining sets O_i, $i = 1, \ldots, 9$, $i \neq k$, are divided into training and validation according to Table 1. This way, across the folders, each object-class appears once and only once as the test set. The resulting numbers of object occurrences to be detected in each of the nine folds are summarized in Table 2, indicating the 3:1 and 10:1 ratios between training/validation and training/test sets, respectively.

Table 1: Separation of objects into folds used to train (Tr), validate (V), and test (Te) the proposed pipeline. The nine object classes are shown in Figure 4.

fold	**O1**	**O3**	**O2**	**O4**	**O5**	**O6**	**O7**	**O8**	**O9**
1	Tr	Tr	Tr	V	V	V	Tr	Tr	Te
2	V	Tr	Tr	Tr	Te	V	V	Tr	Tr
3	Tr	V	V	Tr	Tr	Tr	Te	V	Tr
4	V	Te	V	Tr	Tr	Tr	V	Tr	Tr
5	Tr	V	Te	Tr	V	Tr	Tr	Tr	V
6	V	Tr	V	Te	Tr	V	Tr	Tr	Tr
7	Te	V	Tr	Tr	Tr	Tr	V	Tr	V
8	Tr	V	Tr	V	V	Tr	Tr	Te	Tr
9	Tr	Tr	Tr	V	Tr	Te	Tr	V	V

The network was trained in parts, that is, each module is trained separately one epoch at a time, with the parameters of the other modules frozen,

Table 2: Amount of training, validation and testing samples per fold.

fold	**training samples**	**validation samples**	**testing samples**
1	11,818	4,623	1,206
2	12,498	3,618	2,010
3	12,636	3,618	1,206
4	13,616	3,618	1,206
5	10,862	4,422	1,206
6	14,862	3,618	1,407
7	12,118	3,618	1,206
8	11,680	4,623	1,206
9	12,384	3,819	1,206

for greater stability in the overall training process [47–49], while the ResNet-50 was kept frozen during the whole training. The training algorithms detailed in Algorithm 1, Algorithm 2, and Algorithm 3 are executed in sequence to complete a full epoch. For each fold, 100 training epochs were performed using the Adam optimizer [50] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial values and learning rates were adjusted for each parameter according to Table 3.

Table 3: Parameters and learning rates used to train the proposed pipeline.

Module	**Parameter**	**Initial Value**	**Learning Rate**
DM	reference weights w_{r_i}	1	1×10^{-2}
	target weights w_{t_i}	1	1×10^{-2}
	bias b_i	1×10^{-2}	2×10^{-4}
	combination weights w_i	1	2×10^{-2}
	sigmoid threshold t	-2.15	3×10^{-2}
MM	opening radius R_{o}	1	1×10^{-4}
	closing radius R_{c}	1	13×10^{-3}
CM	sigmoid threshold t_{CM}	2×10^{-2}	1×10^{-4}

As shown in Figure 2, the DM contains 1,025 learnable parameters. The 256 pairs of reference and target weights (w_{r_i}, w_{t_i}) were initialized as $(1, 1)$, and at the end of fold 2 training they assume the values depicted in Figure 5. These weight values are spread across the range $[-4, 5]$, which gives an indi-

cation that the proposed learnable weighted difference is indeed preferred to the mere difference between reference and target feature tensors. The final combination weights w_i follow the same scattered behavior as the weights w_{r_i} and w_{t_i}, spread across the range $[-13, 11]$. At the end of fold 2 training the final bias coefficients b_i were uniformly distributed in the interval $[-0.16, 0.2]$, and the DM threshold was set to $t = -2.86$.

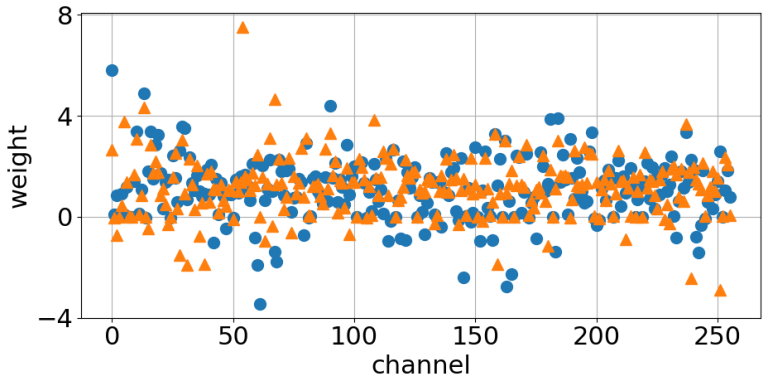


Figure 5: Optimal values of DM parameters for fold 2. The learned reference weights (w_{r_i}) are represented by the blue circles and the target weights (w_{t_i}) are represented by the orange triangles, for $i = 1, 2, \ldots, 256$.

The learnable MM parameters are the two structuring element radii R_o and R_c of the opening and closing operations, respectively, that are applied in sequence to the output image yielded by the DM. Figure 6 shows the evolution of such parameters along the training epochs in fold 2. In this process, the opening radius, that was initialized as $R_\mathrm{o} = 1$, remains at $R_\mathrm{o} = 1$. On the other hand, Figure 6 illustrates that the closing radius R_c evolves along the MM training process, starting from an initial value $R_\mathrm{c} = 1$ and remaining within a small range around $R_\mathrm{c} = 8$ after 50 epochs.

It was observed that the opening radius R_o remains stable during training if its initial value is sufficiently small. This behavior can be explained by the experiment illustrated in Figure 7. In the VDAO dataset, the DM tends to generate anomaly masks containing a few isolated regions of false positive pixels along with the locations of abandoned objects, such as depicted in Figure 7(a). These false positive pixels can be easily removed by an opening with radius $R_\mathrm{o} = 1$, as shown in Figure 7(b). Figures 7(c) and 7(d) show that if one increases the value of the radius R_o to 3 and 5, respectively, then

the output of the MM remains unchanged, that is, the false positives are still removed and there is no visible addition of false negatives inside the object of interest. Indeed, Figures 7(a) to 7(d) show that, for R_o in the range $[1, 5]$, the gradient of any function of the output with respect to R_o is close to zero, which implies that this R_o range corresponds to a local minima plateau. This explains why in our experiments, as long as the initial opening radius is sufficiently small, it tends to remain stable during training.

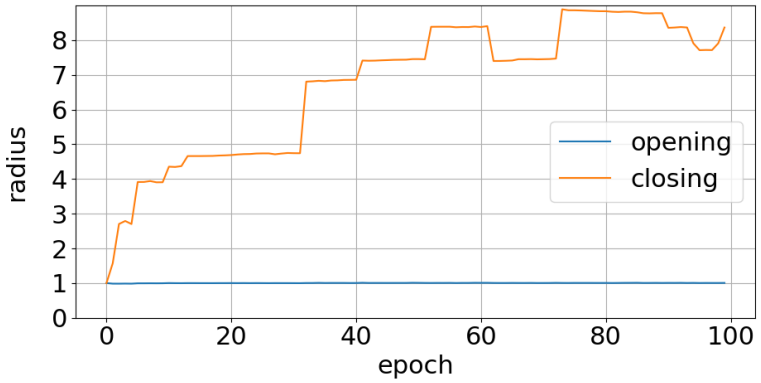


Figure 6: Learning process of MM parameters for fold 2: opening and closing radii.

Figure 8 contains a practical example of the results of the morphological operations performed by the MM using the radii learned according to Figure 6. Note that the proposed learnable operations have the same capabilities of the standard morphological operations, that is, they remove small foreground regions and fill in the object gaps, with the advantage of allowing for parameter adjustment along a gradient-based learning process.

The CM adjusts the threshold value t_{CM} that defines the target frame classification as anomalous or not, depending on the percentage of active (white) pixels in the MM output image. Along the training of fold 2, this value rapidly converges to $t_{CM} = 10^{-3}$, which corresponds to a number of at least 14 active pixels being required to identify the presence of an abandoned object.

The comparative training losses in each pipeline module are shown in Figure 9, illustrating the successful minimization of all loss functions along the learning process of the proposed system.

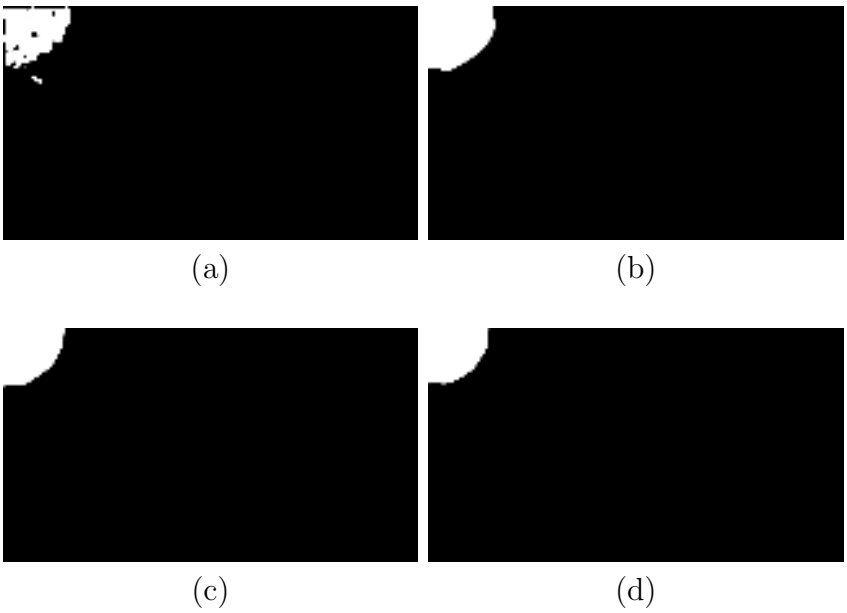


Figure 7: Examples of results obtained by the MM for various values for the radius R_{o} of the opening operation. (a) Anomaly mask before the MM. (b), (c) and (d) represent anomaly masks after the MM with $R_{\mathrm{c}} = 5$ and (b) $R_{\mathrm{o}} = 1$; (c) $R_{\mathrm{o}} = 3$; (d) $R_{\mathrm{o}} = 5$.

5.2. Performance Assessment

In order to assess the performance of the proposed method, henceforth referred to as the TCF-LMO algorithm, we compute both the frame-level and the object-level metrics described in Section 4.2 and compare them to the ones obtained by previous works on the VDAO-200 dataset. Note that, in order to compute object-level metrics for the TCF-LMO, one uses the image provided at the output of the TCM module.

The frame-level $\mathrm{DIS}_{\mathrm{av}}$ and $\mathrm{DIS}_{\mathrm{oa}}$ values on the testing VDAO-200 dataset are shown in Table 4 for the proposed TCF-LMO and other algorithms found in the related literature. From this table, one readily sees that the TCF-LMO surpasses all previous works with respect to both metrics, confirming the effectiveness of our proposed pipeline. A detailed analysis for each of the 59 VDAO-200 videos is shown in Table 6, where bold highlights indicate the best DIS result for each video, 19 of which yielded by the proposed algorithm.

The object-level assessment is summarized in Table 5, where the proposed TCF-LMO algorithm once again outperforms current state-of-the-art methods, confirming its effectiveness also regarding object position and silhouettes.

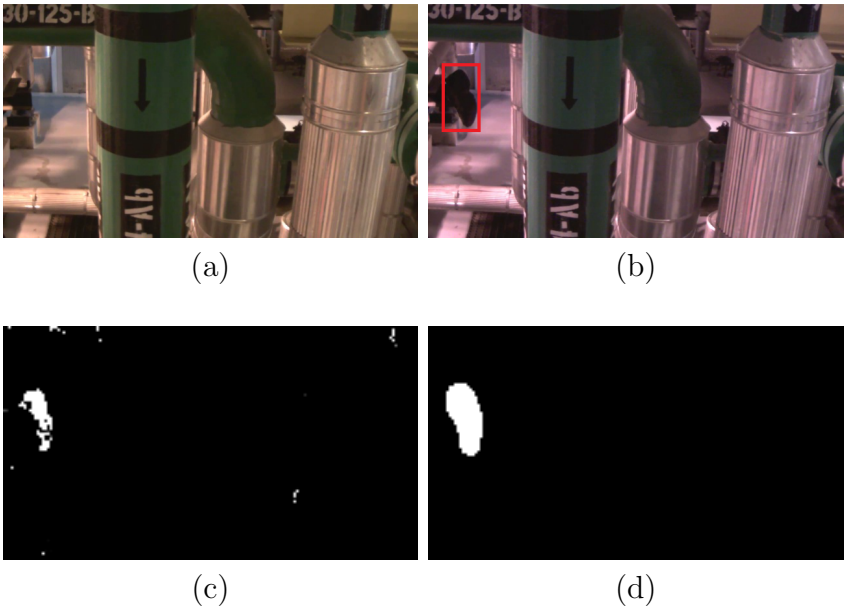


Figure 8: Example of the operation performed by the MM. (a) Reference frame; (b) Target frame containing an anomalous object (red bounding box); (c) Anomaly mask before the MM; (d) Anomaly mask after the MM, where learnable opening and closing operations are performed sequentially.

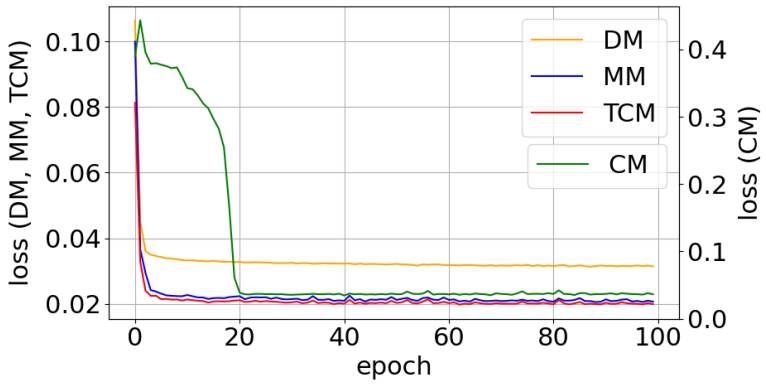


Figure 9: Training losses for all pipeline modules (DM, MM, TCM, and CM) for fold 2.

Table 4: Frame-level results with DIS_{av} and DIS_{oa} in the VDAO-200 compared to other works.

	average			**overall**		
	TPR	**FPR**	**DIS_{av}**	**TPR**	**FPR**	**DIS_{oa}**
DAOMC [10]	0.88	0.39	0.49	0.89	0.42	0.43
ADMULT [16]	0.76	0.36	0.59	0.78	0.39	0.44
MCBS [11]	1.00	0.83	0.83	1.00	0.98	0.98
mcDTSR [17]	0.88	0.25	0.36	0.88	0.28	0.30
CNN+RF [15]	0.74	0.25	0.48	0.75	0.27	0.37
TCF-LMO	0.85	0.18	**0.33**	0.86	0.21	**0.25**

Table 5: Object-level results with DIS_{av} and DIS_{oa} in the VDAO-200 compared to other works.

	average			**overall**		
	TPR	**FPR**	**DIS_{av}**	**TPR**	**FPR**	**DIS_{oa}**
DAOMC [10]	0.81	0.42	0.53	0.82	0.42	0.45
ADMULT [16]	0.70	0.29	0.54	0.72	0.29	0.40
MCBS [11]	0.88	0.83	0.86	0.89	0.83	0.84
mcDTSR [17]	0.86	0.29	0.39	0.86	0.29	0.32
TCF-LMO	0.85	0.22	**0.35**	0.86	0.22	**0.26**

We compare in Table 7 the average processing time on the VDAO-200 dataset obtained by the TCF-LMO and all other methods found in the literature. The tests were performed on a computer with CPU Intel Core i7-4790K @ 4.00GHz, 32GB of RAM, and a discrete NVIDIA Geforce GTX Titan XP with 12GB of VRAM. The proposed method and the CNN+RF were simulated in inference mode, and the training time for the modules is not considered. One can readily see that the proposed method is 1.8 times faster than the current fastest method (CNN+RF).

We can attribute this behavior to the proposed method being designed with a reduced number of parameters for the limited samples scenario. A direct comparison of the number of parameters can be made with the CNN+MLP model also proposed by Afonso et al. [15], since both use the same features from ResNet 50. The proposed TCF-LMO method obtains a fast and reliable estimate with a total of only 1,028 trainable parameters, while the CNN+MLP requires a total of 90,475 parameters.

Table 6: Detailed frame-level analysis for all 59 VDAO-200 videos of proposed TCF-LMO method compared to other works found in the literature.

	DAOMC [10]			ADMULT [16]			MCBS [11]			mcDTSR [17]			CNN + RF [15]			**TCF-LMO**		
video	TPR	FPR	DIS	TPR	FPR	DIS	TPR	FPR	DIS	TPR	FPR	DIS	TPR	FPR	DIS	TPR	FPR	DIS
1	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.44	0.00	0.56	1.00	0.00	**0.00**
2	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.73	0.00	0.27	0.10	0.18	0.92	0.72	0.00	0.28
3	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.91	0.00	0.09	1.00	0.00	**0.00**
4	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.67	0.00	0.33	1.00	0.00	**0.00**
5	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.89	0.00	0.11	1.00	0.00	**0.00**
6	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	1.00	1.00	1.00	0.00	**0.00**
7	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.82	0.33	0.38	1.00	0.00	**0.00**
8	0.76	0.71	0.75	0.57	0.00	0.43	1.00	0.99	0.99	1.00	0.38	**0.38**	0.95	0.41	0.42	0.45	0.00	0.55
9	0.67	0.00	0.33	0.60	0.00	0.40	1.00	1.00	1.00	0.97	0.00	**0.03**	0.96	0.00	0.04	0.96	0.00	0.04
10	0.89	0.00	0.11	0.69	0.00	0.31	1.00	1.00	1.00	0.96	0.00	**0.04**	0.63	0.33	0.50	0.92	0.00	0.08
11	0.82	0.32	0.37	1.00	1.00	1.00	1.00	0.99	0.99	0.91	0.48	0.49	0.11	0.00	0.89	0.91	0.00	**0.09**
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.90	0.00	**0.10**	1.00	0.23	0.23	1.00	1.00	1.00
13	0.84	0.00	0.16	0.65	0.00	0.35	0.88	0.81	0.82	1.00	0.00	**0.00**	0.92	0.00	0.08	0.90	0.00	0.10
14	0.93	0.00	**0.07**	1.00	0.52	0.52	1.00	0.93	0.93	0.88	0.00	0.12	0.91	1.00	1.00	0.80	0.00	0.20
15	1.00	1.00	1.00	0.63	0.00	0.37	1.00	1.00	1.00	1.00	0.12	0.12	0.97	0.00	**0.03**	0.91	0.00	0.09
16	0.00	0.00	1.00	0.00	0.00	1.00	1.00	0.98	0.98	0.89	0.31	**0.33**	0.55	0.08	0.46	0.00	0.00	1.00
17	0.81	0.70	0.72	0.79	1.00	1.02	0.99	1.00	1.00	0.96	0.00	**0.04**	0.92	0.00	0.08	0.95	0.03	0.06
18	0.43	0.00	0.57	0.26	0.13	0.76	1.00	1.00	1.00	0.55	0.00	**0.45**	0.00	0.00	1.00	0.00	0.00	1.00
19	0.89	0.00	0.11	0.74	0.00	0.26	1.00	0.94	0.94	0.95	0.00	**0.05**	0.66	0.00	0.34	0.86	0.00	0.14
20	1.00	1.00	1.00	0.00	0.00	1.00	0.99	1.00	1.00	1.00	0.91	**0.91**	1.00	1.00	1.00	1.00	1.00	1.00
21	1.00	0.30	0.30	1.00	1.00	1.00	1.00	0.69	0.69	1.00	0.09	**0.09**	0.99	0.14	0.14	0.55	0.00	0.45
22	0.94	0.21	0.22	1.00	1.00	1.00	1.00	0.97	0.97	1.00	0.21	0.21	1.00	1.00	1.00	0.93	0.00	**0.07**
23	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.72	0.17	0.32	1.00	0.00	**0.00**	1.00	1.00	1.00
24	0.97	0.18	**0.18**	0.00	0.00	1.00	0.99	1.00	1.00	0.12	0.00	0.88	0.09	0.00	0.91	0.00	0.00	1.00
25	0.58	0.00	0.42	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.54	0.00	0.46	0.16	0.00	0.84	0.55	0.00	0.45
26	0.93	0.00	**0.07**	0.68	0.00	0.32	1.00	0.98	0.98	1.00	0.23	0.23	1.00	0.10	0.10	1.00	0.09	0.09
27	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.57	0.00	0.43	1.00	0.00	**0.00**
28	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.00	**0.00**	0.26	0.00	0.74	1.00	1.00	1.00	1.00	0.00	**0.00**
29	0.76	0.00	0.24	0.69	0.00	0.31	1.00	1.00	1.00	0.00	0.00	1.00	1.00	0.00	**0.00**	0.81	0.00	0.19
30	0.80	0.00	0.20	0.56	0.00	0.44	1.00	0.98	0.98	1.00	0.16	0.16	1.00	0.00	**0.00**	1.00	0.09	0.09
31	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.87	0.00	**0.13**	1.00	1.00	1.00
32	0.83	0.00	0.17	0.32	0.00	0.68	1.00	1.00	1.00	0.99	0.06	**0.06**	0.71	0.00	0.29	1.00	0.06	**0.06**
33	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.63	0.13	**0.39**	1.00	1.00	1.00
34	0.70	0.00	0.30	0.56	0.00	0.44	1.00	0.98	0.98	0.97	0.09	**0.10**	0.63	0.00	0.37	0.88	0.00	0.13
35	1.00	0.20	0.20	0.63	0.00	0.37	1.00	0.93	0.93	0.96	0.00	0.04	1.00	0.00	**0.00**	0.90	0.00	0.10
36	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.27	**0.27**	0.98	0.90	0.90	1.00	1.00	1.00
37	0.93	0.00	0.07	0.93	0.00	0.07	1.00	1.00	1.00	0.97	0.00	0.03	0.55	0.00	0.45	0.98	0.00	**0.02**
38	0.76	0.13	0.28	0.47	0.00	0.53	1.00	0.97	0.97	0.76	0.03	0.25	0.22	1.00	1.27	0.98	0.00	**0.02**
39	1.00	1.00	**1.00**	1.00	1.00	**1.00**	1.00	1.00	**1.00**	1.00	1.00	**1.00**	0.00	0.10	1.00	1.00	1.00	**1.00**
40	1.00	1.00	1.00	1.00	0.59	0.59	1.00	1.00	1.00	1.00	1.00	1.00	0.83	0.00	0.17	0.91	0.00	**0.09**
41	1.00	0.95	0.95	1.00	1.00	1.00	1.00	0.97	0.97	1.00	0.09	**0.09**	0.79	0.17	0.26	1.00	0.38	0.38
42	1.00	1.00	1.00	0.50	0.00	0.50	1.00	0.97	0.97	0.99	0.00	0.01	1.00	0.00	**0.00**	0.93	0.00	0.07
43	0.14	0.00	0.86	0.00	0.00	1.00	1.00	1.00	1.00	0.93	0.26	**0.27**	0.96	0.59	0.59	0.00	0.00	1.00
44	0.78	0.38	0.44	0.63	0.00	0.37	1.00	1.00	1.00	0.95	0.00	**0.05**	0.75	0.00	0.25	0.81	0.00	0.19
45	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.37	1.00	1.18	0.88	0.14	**0.18**	1.00	1.00	1.00
46	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.90	0.00	**0.10**	0.78	0.00	0.22	0.90	0.00	**0.10**
47	0.93	0.00	0.07	0.91	0.00	0.09	1.00	1.00	1.00	0.97	0.00	**0.03**	1.00	1.00	1.00	0.87	0.00	0.13
48	0.72	0.00	0.28	0.42	0.00	0.58	1.00	1.00	1.00	0.98	0.00	0.02	1.00	0.00	**0.00**	0.98	0.00	0.02
49	1.00	0.20	0.20	0.93	0.00	**0.07**	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	0.48	0.00	0.52
50	0.97	0.00	**0.03**	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.00	**0.03**	0.96	0.58	0.58	0.93	0.00	0.07
51	0.96	0.86	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.07	**0.07**	1.00	1.00	1.00
52	0.84	0.82	0.83	1.00	1.00	1.00	1.00	1.00	1.00	0.74	0.00	**0.26**	1.00	1.00	1.00	1.00	1.00	1.00
53	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	0.00	**0.20**
54	0.85	0.00	0.15	0.50	0.00	0.50	1.00	1.00	1.00	1.00	0.00	**0.00**	1.00	0.00	**0.00**	1.00	0.02	0.02
55	0.79	0.67	0.70	0.50	0.00	0.50	1.00	1.00	1.00	0.71	0.00	**0.29**	0.00	0.00	1.00	0.70	0.00	0.30
56	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.00	**0.03**	0.98	0.18	0.18
57	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	1.00	0.97	0.00	**0.03**
58	0.62	0.00	0.38	0.19	0.00	0.81	1.00	1.00	1.00	0.88	0.00	**0.12**	0.88	0.00	**0.12**	0.87	0.00	0.13
59	1.00	0.49	0.49	0.54	0.00	0.46	1.00	1.00	1.00	0.62	0.01	0.38	0.00	0.00	1.00	1.00	0.01	**0.01**
Average	0.88	0.39	0.49	0.76	0.36	0.59	1.00	0.83	0.83	0.88	0.25	0.36	0.74	0.25	0.48	0.85	0.18	**0.33**

Table 7: Average processing time (in seconds) on each video of the VDAO-200 videos for the proposed TCF-LMO compared to other methods.

DAOMC	ADMULT	MCBS	mcDTSR	CNN + RF	**TCF-LMO**
284	114	50 543	1 050	6.5	**3.7**

Figure 10 shows examples of successful anomaly masks generated by the TCM for the black backpack, white jar, dark blue box and shoe (see Figure 4). One can see that the silhouettes of the anomalous objects are well represented with the proposed method, which is noteworthy for the example in Figures 10(c) and (d) since it contains a black object in a shadowy region of the scene.

Figure 11 shows examples of failure cases, where the individual DIS obtained in Table 6 was close to 1. One can categorize such results in three cases: small non-salient objects, such as the example in Figure 11(a); frames where the camera is slightly rotated with respect to the equivalent position in the reference frame, which might occur on the position shown in Figure 11(c) due to problems on the video acquisition; and objects that cast shadows on the background or reflection on metallic surfaces, as shown in Figure 11(e), where the shadow is computed as a false positive since it does not appear in the ground truth annotation (note that this case should not be considered as a failure, since a shadow introduces a change in the video).

Results for all the videos on the VDAO-200, as well as all codes related to the TCF-LMO are available at `https://github.com/rafaelpadilla/TCF-LMO`.

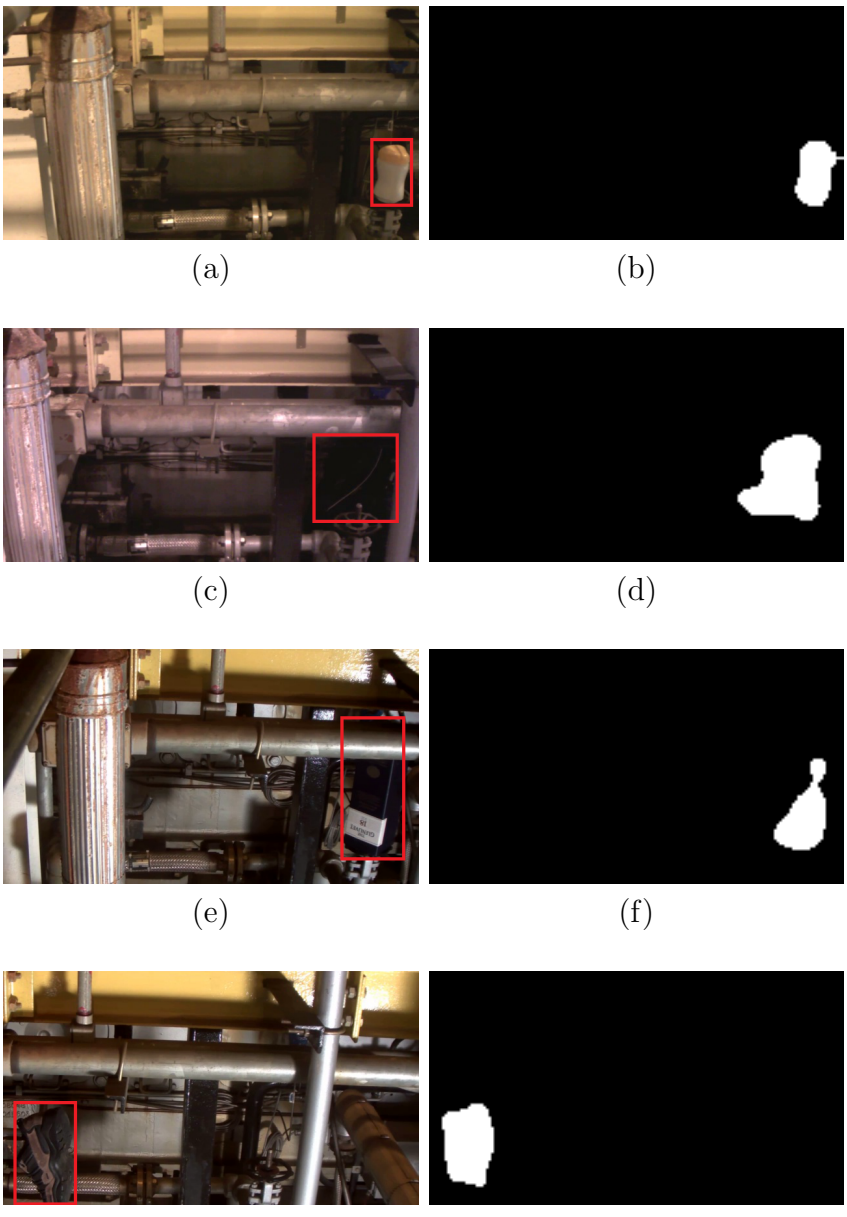


Figure 10: Examples of successful anomaly masks determined by the TCM output image. (a) Target frame containing a white jar (red bounding box); (b) Output image; (c) Target frame containing a black backpack (red bounding box); (d) Output image; (e) Target frame containing a dark blue box (red bounding box); (f) Output image; (g) Target frame containing a shoe (red bounding box); (h) Output image.

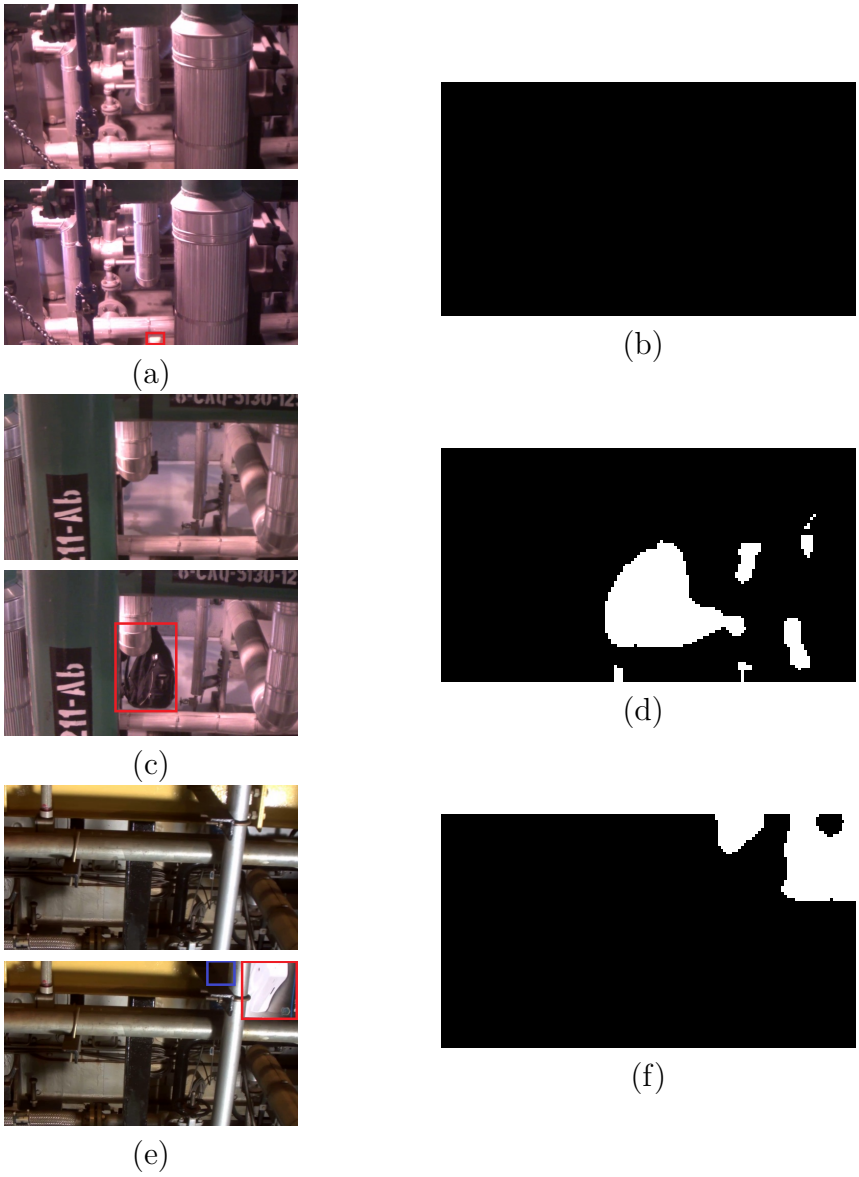


Figure 11: Examples of failure cases observed for the TCM output image. (a) Reference (above) and target (below) containing a small non-salient white jar (red bounding box); (b) Output image; (c) Reference (above) and target (below) that is misaligned to the reference frame (note the characters occlusion in the image borders and the rotation in the vertical pipes), containing a black backpack (red bounding box); (d) Output image; (e) Reference (above) and target (below) containing a camera box (red bounding box) that casts a shadow (blue bounding box) on the background; (f) Output image.

5.3. Ablation Studies

To evaluate the effectiveness of each module in the proposed TCF-LMO framework, we perform experiments with different system configurations, including (✓) or not including (×) some of its modules, as given in Table 8. When removing the proposed DM, we replace it by the Euclidian distance between reference and target feature tensors employed by Afonso et al. [15]. A version without the TCM does not impose any temporal consistency on the results, and a version without the MM does not apply any morphological post-processing on the results.

Table 8 shows the frame-level DIS_{av} and DIS_{oa} values obtained on the testing VDAO-200 dataset using the different versions of the TCF-LMO pipeline. Such results show how the proposed low-complexity DM considerably enhances overall detection performance, indicating its better ability to combine reference and target features despite its reduced number of parameters that are trainable even on constrained datasets.

Table 8: Ablation study of the proposed method, with frame-level results obtained for different versions of the proposed pipeline, where we maintained (✓) or removed (×) the DM, TCM and/or MM.

TCF-LMO			**average**			**overall**		
DM	**MM**	**TCM**	**TPR**	**FPR**	$\mathbf{DIS_{av}}$	**TPR**	**FPR**	$\mathbf{DIS_{oa}}$
✓	✓	✓	0.85	0.18	**0.33**	0.86	0.21	**0.25**
×	✓	✓	0.75	0.37	0.61	0.75	0.44	0.51
✓	✓	×	0.83	0.18	0.34	0.85	0.20	**0.25**
✓	×	✓	0.85	0.23	0.37	0.86	0.28	0.31
✓	×	×	0.86	0.22	0.35	0.87	0.25	0.28

Table 8 results also indicate how the learnable MM leads to a consistent reduction of the DIS values demonstrating its practical effectiveness. As can also be seen from Fig 6, the MM is able to learn the optimal radius of a structuring element that is used to remove small false positive regions while keeping most true positives intact.

The experiments evaluating the TCM role show that this module does not significantly improve the detection performance in any configuration. This could indicate that the proposed DM produces a robust enough detection mask with a small amount of consistent false detections that are not easily removed by a voting procedure (e.g. false positives due to the shadow of the

abandoned objects). Nonetheless, the best performance was obtained with the complete architecture including the TCM. It should be noted, however, that in order to learn the voting-window length, batches of consecutive frames are required by the TCM, increasing the time and memory consumption. Considering the above, we conclude that both architectures have benefits: the complete TCF-LMO for the general case, and a version without the TCM for resource-limited applications.

6. Conclusion

This work proposes a new pipeline to perform video change detection that computes the dissimilarity between features extracted from twin deep convolutional networks and processes it using learnable morphological operations. The so-called TCF-LMO system contains a total of 1,028 trainable parameters distributed across four processing modules: a low-complexity dissimilarity module (DM), a learnable morphology module (MM), a temporal consistency module (TCM), and a classification module (CM). Each module was designed to perform specific operations, which, when applied in sequence, aim to contribute to the correct frame classification and change detection. The pipeline was developed considering the limitations in the size of the VDAO dataset, and can in principle be adapted to work in any change detection application with a small training database.

System development considers a training/validation/testing scheme that takes maximum advantage of the scarce data resources while avoiding data contamination across different learning stages. A loss function analysis depicts a successful training process, and final test results showed that the proposed scheme was able to outperform all previous methods assessed with the VDAO dataset in the literature. Such results indicate the successful ability of the proposed TCF-LMO system to detect, at both frame- and object-levels, changes represented by abandoned objects in videos acquired using moving cameras.

An innovative dissimilarity module is proposed that fuses the feature tensors of reference and target frames. The proposed module has a simple structure, enabling an effective training process on databases with a scarce number of samples. The ResNet-50 features were chosen in accordance to previous works for a direct comparison and were not retrained, but the proposed DM architecture can be adapted to any given backbone. Novel learnable morphological operations are introduced allowing one to optimize their parameters

(structuring element radii) during a gradient-descent training process. The MCC metric has been slightly modified to avoid degenerate cases and is used as the guiding loss function for the DM, MM, and TCM training processes. Ablation studies performed with variations of the proposed architecture confirm the contribution of each newly proposed module on the overall system performance.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ). Most of the experimental results reported in this work were obtained with a Titan X Pascal board gently donated by the NVIDIA Corporation.

References

[1] M.-N. Chapel, T. Bouwmans, Moving objects detection with a moving camera: A comprehensive review, Computer Science Review 38 (2020) 100310.

[2] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, Journal of ACM Computing Surveys (2009) 1–58.

[3] T. D. Räty, Survey on contemporary remote surveillance systems for public safety, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40 (5) (2010) 493–515.

[4] H.-C. Shin, J.-Y. Lee, Pedestrian video data abstraction and classification for surveillance system, in: Proceedings of the IEEE International Conference on Information and Communication Technology Convergence, Jeju, South Korea, 2018, pp. 1476–1478.

[5] A. F. da Silva, L. A. Thomaz, G. Carvalho, M. T. Nakahata, E. Jardim, J. F. L. de Oliveira, E. A. B. da Silva, S. L. Netto, G. Freitas, R. R. Costa, An annotated video database for abandoned-object detection in a cluttered environment, in: Proceedings of the International Telecommunications Symposium, São Paulo, Brazil, 2014, pp. 1–5.

[6] A. Romanoni, M. Matteucci, D. G. Sorrenti, Background subtraction by combining temporal and spatio-temporal histograms in the presence of camera movement, Machine Vision and Applications 25 (6) (2014) 1573–1584.

[7] L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, S. L. Netto, H. Krim, Detection of abandoned objects using robust subspace recovery with intrinsic video alignment, in: IEEE International Symposium on Circuits and Systems, 2017, pp. 1–4.

[8] E. Jardim, X. Bian, E. A. B. da Silva, S. L. Netto, H. Krim, On the detection of abandoned objects with a moving camera using robust subspace recovery and sparse representation, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, South Brisbane, Australia, 2015, pp. 1295–1299.

[9] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, USA, 2010, pp. 1975–1981.

[10] H. Kong, J.-Y. Audibert, J. Ponce, Detecting abandoned objects with a moving camera, IEEE Transactions on Image Processing 19 (8) (2010) 2201–2210.

[11] H. Mukojima, D. Deguchi, Y. Kawanishi, I. Ide, H. Murase, M. Ukai, N. Nagamine, R. Nakasone, Moving camera background-subtraction for obstacle detection on railway tracks, in: Proceedings of the IEEE International Conference on Image Processing, Phoenix, USA, 2016, pp. 3967–3971.

[12] C. Zhao, A. Sain, Y. Qu, Y. Ge, H. Hu, Background subtraction based on integration of alternative cues in freely moving camera, IEEE Transactions on Circuits and Systems for Video Technology 29 (7) (2019) 1933–1945.

[13] M. T. Nakahata, L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, S. L. Netto, Anomaly detection with a moving camera using spatio-temporal codebooks, Multidimensional Systems and Signal Processing 29 (3) (2018) 1025–1054.

[14] L. A. Thomaz, E. Jardim, A. F. da Silva, E. A. B. da Silva, S. L. Netto, H. Krim, Anomaly detection in moving-camera video sequences using principal subspace analysis, IEEE Transactions on Circuits and Systems 65 (3) (2017) 1003–1015.

[15] B. M. Afonso, L. P. Cinelli, L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, S. L. Netto, Moving-camera video surveillance in cluttered environments using deep features, in: Proceedings of the IEEE International Conference on Image Processing, Athens, Greece, 2018, pp. 2296–2300.

[16] G. H. F. Carvalho, L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, S. L. Netto, Anomaly detection with a moving camera using multiscale video analysis, Multidimensional Systems and Signal Processing 30 (1) (2019) 311–342.

[17] E. Jardim, L. Thomaz, E. A. B. da Silva, S. L. Netto, Domain-transformable sparse representation for anomaly detection in moving-camera videos, in: IEEE Transactions on Image Processing, Vol. 29, 2020, pp. 1329–1343.

[18] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[19] S. Shahinfar, P. Meek, G. Falzon, "How many images do i need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring, Ecological Informatics 57 (2020) 101085.

[20] M. Mandal, V. Dhar, A. Mishra, S. K. Vipparthi, M. Abdel-Mottaleb, 3DCD: Scene independent end-to-end spatio-temporal feature learning framework for change detection in unseen videos, IEEE Transactions on Image Processing 30 (2020) 546–558.

[21] A. Dore, M. Soto, C. S. Regazzoni, Bayesian tracking for video analytics, IEEE Signal Processing Magazine 27 (5) (2010) 46–55.

[22] B. N. Subudhi, P. K. Nanda, A. Ghosh, A change information based fast algorithm for video object detection and tracking, IEEE Transactions on Circuits and Systems for Video Technology 21 (2011).

[23] V. Saligrama, J. Konrad, P.-M. Jodoin, Video anomaly identification, IEEE Signal Processing Magazine 27 (5) (2010) 18–33.

[24] L. Cheng, M. Gong, D. Schuurmans, T. Caelli, Real-time discriminative background subtraction, IEEE Transactions on Image Processing 20 (5) (2010) 1401–1414.

[25] P.-M. Jodoin, V. Saligrama, J. Konrad, Behavior subtraction, IEEE Transactions on Image Processing 21 (9) (2012) 4244–4255.

[26] T. P. Nguyen, C. C. Pham, S. V.-U. Ha, J. W. Jeon, Change detection by training a triplet network for motion feature extraction, IEEE Transactions on Circuits and Systems for Video Technology 29 (2) (2019) 433–446.

[27] H. Sajid, S. ching S. Cheung, N. Jacobs, Appearance based background subtraction for PTZ cameras, Signal Processing: Image Communication 47 (2016) 417–425.

[28] Y. Tomioka, A. Takara, H. Kitazawa, Generation of an optimum patrol course for mobile surveillance camera, IEEE Transactions on Circuits and Systems for Video Technology 22 (2) (2011) 216–224.

[29] H. Sajid, S.-C. S. Cheung, N. Jacobs, Motion and appearance based background subtraction for freely moving cameras, Signal Processing: Image Communication 75 (2019) 11–21.

[30] W.-C. Hu, C.-H. Chen, T.-Y. Chen, D.-Y. Huang, Z.-C. Wu, Moving object detection and tracking from video captured by moving camera, Journal of Visual Communication and Image Representation 30 (2015) 164–180.

[31] M. J. Roshtkhari, M. D. Levine, An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions, Journal of Computer Vision and Image Understanding 117 (10) (2013) 1436–1452.

[32] X. Bian, H. Krim, Bi-sparsity pursuit for robust subspace recovery, in: Proceedings of the IEEE International Conference on Image Processing, Quebec City, Canada, 2015, pp. 3535–3539.

[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp. 770–778.

[34] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018, pp. 7132–7141.

[35] X. Jin, Y. Xie, X.-S. Wei, B.-R. Zhao, Z.-M. Chen, X. Tan, Delving deep into spatial pooling for squeeze-and-excitation networks, Pattern Recognition 121 (2022) 108159.

[36] B. Hu, X. Wang, W. Yu, Joint weakly and fully supervised learning for surface defect segmentation from images, Signal Processing: Image Communication 107 (2022) 116807.

[37] Y. Wu, D. Zhang, F. Yin, Y. Zhang, Salient object detection based on global to local visual search guidance, Signal Processing: Image Communication 102 (2022) 116618.

[38] V. Kumar, R. S. Singh, Y. Dua, Morphologically dilated convolutional neural network for hyperspectral image classification, Signal Processing: Image Communication 101 (2022) 116549.

[39] Y. Hao, L. Dong, X. Liao, J. Liang, L. Wang, B. Wang, A novel clustering algorithm based on mathematical morphology for wind power generation prediction, Renewable Energy 136 (2019) 572–585.

[40] L. Liu, Z. Jia, J. Yang, N. K. Kasabov, Sar image change detection based on mathematical morphology and the k-means clustering algorithm, IEEE Access 7 (2019) 43970–43978.

[41] S. Boughorbel, F. Jarray, M. El-Anbari, Optimal classifier for imbalanced data using Matthews correlation coefficient metric, PloS ONE 12 (6) (2017) e0177678.

[42] D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, BMC Genomics 21 (1) (2020) 1–13.

[43] D. Chicco, N. Tötsch, G. Jurman, The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation, BioData Mining 14 (1) (2021) 1–22.

[44] K. Abhishek, G. Hamarneh, Matthews correlation coefficient loss for deep convolutional networks: Application to skin lesion segmentation, in: Proceedings of the IEEE International Symposium on Biomedical Imaging, Nice, France, 2021, pp. 225–229.

[45] VDAO: Video database of abandoned objects in a cluttered industrial environment, `http://www.smt.ufrj.br/~tvdigital/database/objects`, accessed: 2023-03-06 (2014).

[46] VDAO-200: 200-frame excerpts from VDAO database, `http://www.smt.ufrj.br/~tvdigital/database/research`, accessed: 2023-03-06 (2017).

[47] Y. Hu, A. Huber, J. Anumula, S.-C. Liu, Overcoming the vanishing gradient problem in plain recurrent networks, arXiv preprint arXiv:1801.06105 (2018).

[48] M. Roodschild, J. G. Sardiñas, A. Will, A new approach for the vanishing gradient problem on sigmoid activation, Progress in Artificial Intelligence 9 (4) (2020) 351–360.

[49] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Advances in Neural Information Processing Systems 19 (2007) 153.

[50] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, San Diego, USA, 2015.